Hyperspectral TV Denoising and Inpainting

Vicknesh Selvam
August 17, 2007
UCLA Department of Mathematics
Project supervised by Prof. Andrea Bertozzi and Todd Wittman

Background:

Total Variation (TV) denoising\(^1\) has been used in the past for spatial denoising, i.e. denoising within a single frame. This method is extremely effective in reducing the minimization of noise while preserving the information present in the image. The energy minimization of a grayscale image \(u : \Omega \rightarrow \mathbb{R} \subseteq \mathbb{R}^2\) is:

\[
\min_{u} E[u | f] = \lambda \int_{\Omega} (u - f)^2 \, dx + \int_{\Omega} \sqrt{(u_x)^2 + (u_y)^2} \, dx
\]

where \(f\) is the original image.

The first term on the right hand side is commonly referred to as the fidelity function, which is responsible for preserving the information within an image. The second term is the regularization term that is responsible for smoothing, and thus noise reduction.

In hyperspectral imaging, we no longer deal with individual images, but instead deal with a set of images on different wavelengths. Thus, \(\Omega \subseteq \mathbb{R}^3\). The image is stored as a ‘volume,’ with the \(x\) and \(y\) coordinates as the spatial coordinates of a frame, and the \(z\) coordinate represents the spectral wavelength, as shown in Figure 1. Naturally, This should not be denoised as a volume, since the spectral correlation is stronger than each pixel’s spatial correlation. Thus, a third term is added to the energy minimization term, to create the following equation:

\[
\min_{u} E[u | f] = \lambda \int_{\Omega} (u - f)^2 \, dx + \int_{\Omega} \sqrt{(u_x)^2 + (u_y)^2} \, dx + \beta \int_{\Omega} (u_z)^2 \, dx
\]

The Euler-Lagrange equation for this energy is:

\[
\frac{\partial u}{\partial t} = -2\lambda (u - f) + \frac{\partial}{\partial x} \left( \frac{u_x}{\sqrt{(u_x)^2 + (u_y)^2}} \right) + \frac{\partial}{\partial y} \left( \frac{u_y}{\sqrt{(u_x)^2 + (u_y)^2}} \right) + 2\beta u_z
\]

In the above equations, \(\lambda\) and \(\beta\) are parameters used to vary the importance of the fidelity function and the spectral correlation function.

Fig 1: Illustration of xyz coordinated for hyperspectral image.

---

\(^1\) Rubin-Osher-Fatemi, 1992
Numerical Implementation:

Denoising is simply the removal of noise points from an image by smoothing it out with respect to its surrounding pixels. Since the noise is difficult to see in the original data, we add Gaussian noise to the images, with mean 0 and variance dependent on the desired outcome. This allows us to see a difference between the original image and the final product. In this report, ‘light noise’ would refer to variance 0.01 and ‘heavy noise’ would refer to variance 0.1. We evolve the PDE using the explicit forward Euler method. For these experiments, while performing pure denoising on noisy images, we fix $\lambda = 1$.

Inpainting, on the other hand, is the act of reconstructing a damaged portion of the data using all available information around it. Thus, the fidelity function has greater importance, since we want to preserve the images that we know are reliable. Thus, $\lambda = 5$ for these experiments. On the other hand, $\lambda = 0$ at all points that we intend to inpaint, removing the fidelity function at these points altogether.

Two data sets, urban and hyperion, are used to show the effectiveness of these methods.

The runtime of the algorithm with respect to the number of frames $N$ seems to be $O(N)$, in the absence of memory concerns (which essentially means approximately 2GB of RAM for 210 bands). Using visual comparisons, the images also seem to reach a steady state after about 7 iterations, with $T = 0.7$ and $dt = 0.1$.

<table>
<thead>
<tr>
<th>Number of Bands</th>
<th>Runtime (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>17.768</td>
</tr>
<tr>
<td>10</td>
<td>35.182</td>
</tr>
<tr>
<td>20</td>
<td>73.087</td>
</tr>
<tr>
<td>40</td>
<td>158.066</td>
</tr>
<tr>
<td>210</td>
<td>786.32</td>
</tr>
</tbody>
</table>

Denoising Results:

By comparing the pictures for $\beta = 0$ and $\beta = 1$, it is clear that the spectral correlation improves denoising. However, the algorithm destabilizes for high values of $\beta$, as can be seen when $\beta = 2.2$. 

2.1: Original image  
2.2: Heavy Noise  
2.3: $\beta = 0$

2.4: $\beta = 1$  
2.5: $\beta = 2.2$

Fig 2: Frame 16 out of 70 of the Hyperion data set with heavy noise added

3.1: Original image  
3.2: Heavy noise  
3.3: $\beta = 0$

3.4: $\beta = 1$  
3.6: $\beta = 2.2$

Fig 3: Frame 16 of 30 of the Urban data set with heavy noise added
Inpainting Results:

When inpainting, the importance of the spectral correlation term is even more evident. In Figure 4 below, using the urban data set, a piece is removed and inpainting is attempted using different values of $\beta$.

4.1: Piece removed  
4.2: $\beta=0$  
4.3: $\beta=2$

Fig 4: Piece of Frame 6 of 10 removed

These results were produced after 7 iterations ($dt=0.1$, $T=0.7$), to show the comparison between the two methods. Had the experiment in Figure 4.2 been run to completion, the result would mostly be a uniformly grey patch replacing the static. However, after just 7 iterations for $\beta=2$, a reasonable restoration of the original image has been achieved. Figures 5 and 6 will show the same method applied to both data sets, where entire frames are removed.

5.1: Original Frame 14  
5.2: $\beta=0$  
5.3: $\beta=2$

Fig 5: Frame 14 of 18 removed from Hyperion

6.1: Original Frame 6  
6.2: $\beta=0$  
6.3: $\beta=2$

Fig 6: Frame 6 of 10 removed from Urban
A concern is that inpainting can be done simply by taking an average of the adjacent frames, or taking a weighted average in the case of multiple damaged channels. However, TV inpainting has an advantage in that it not only reconstructs the image, it also performs TV denoising on the other channels. Fig 7 below shows a repeat of the experiment in Fig 6, but with heavy noise added to the images. Fig 7a shows channels 5-7, where 6 is the removed channel, and 7b is channel 6 after inpainting. Fig 7c is simply an average of the two adjacent channels.

![Fig 7: TV denoising-inpainting vs. channel averaging.](image)

**Fig 7:** TV denoising-inpainting vs. channel averaging.

**Multi-Channel Inpainting:**

The same method can be used to inpaint multiple channels. However, since each iteration only takes into account the adjacent frames, inpainting multiple channels would take considerably longer. The urban data set has several sets of consecutive frames that are damaged or uncalibrated. Particularly, frames 101-111 are marked as damaged frames. Figure 8 shows bands 99-113. Figure 9 is the same range, after all 210 bands were used in inpainting. The algorithm was run for 100 iterations to ensure that each image reaches a steady state. Figure 9 is the result after 40 iterations. At 100 iterations, however, even the ‘good’ images get smoothed out to a point where they are slightly blurred. Even comparing Fig 8 and 9, Frame 103 (top right) gets blurred somewhat. However, we must note that it was classified as a damaged frame, and that the frames in this section that are classified as undamaged (99,100,112,113) are preserved.
Fig 8: Frames 99-113 of Urban

Fig 9: Frames 99-113 of Urban after inpainting (dt=0.1, T=4)